Interrupt Controller

Table of Contents

[Overview 2](#_Toc353131561)

[Registers 2](#_Toc353131562)

[INTENABLE 2](#_Toc353131563)

[INTREQUEST 2](#_Toc353131564)

# Overview

The Interrupt Controller is responsible for generating interrupt signals to the CPU. It’s also responsible for giving the CPU a way to clear an interrupt request, once it has been handled.

There are a number of possible interrupt sources. VBlank, Timers, and potentially a specific scanline being visited, could all be interrupt sources. Once an interrupt source sets its’ IRQ flag, the VideoChip asserts the /INT line, which goes directly to the CPU. The assertion remains high until the CPU instructs the videochip to clear the IRQ. The videochip will only negate the /INT signal if other interrupt sources are not also requesting action.

# Registers

## INTENABLE

76543210  
MVKxABCD

M: Master Enable (0 = all other bits in this register are treated as if they are 0)  
V: 1 = VBlank interrupts are enabled  
K: 1 = Copper interrupts are enabled  
ABCD: 4 timers, named A, B, C, and D, can all produce interrupts at different speeds.

## INTREQUEST

This register shares the same format as the INTENABLE register. However, this register allows the CPU to determine WHY an interrupt has just occurred. Check for a 1 for each device which you suspect could have generated the interrupt, to see if it indeed needs to be serviced.

Writing to this register is unique, in that writing a 1 to any particular bit(s) causes the bit(s) to be cleared.

Even if the Interrupt Enable flag is turned off for a particular device, this register will still reflect whether an event for that device has occurred.